Style Neophile: Constantly Seeking Novel Styles for Domain Generalization

Juwon Kang[1] Sohyun Lee[2] Namyup Kim[1] Suha Kwak[1,2]
Dept. of CSE, POSTECH[1], Graduate School of AI, POSTECH[2]
{gjw0917, lshig96, namyup, suha.kwak}@postech.ac.kr

Abstract

This paper studies domain generalization via domain-invariant representation learning. Existing methods in this direction suppose that a domain can be characterized by styles of its images, and train a network using style-augmented data so that the network is not biased to particular style distributions. However, these methods are restricted to a finite set of styles since they obtain styles for augmentation from a fixed set of external images or by interpolating those of training data. To address this limitation and maximize the benefit of style augmentation, we propose a new method that synthesizes novel styles constantly during training. Our method manages multiple queues to store styles that have been observed so far, and synthesizes novel styles whose distribution is distinct from the distribution of styles in the queues. The style synthesis process is formulated as a monotone submodular optimization, thus can be conducted efficiently by a greedy algorithm. Extensive experiments on four public benchmarks demonstrate that the proposed method is capable of achieving state-of-the-art domain generalization performance.

1. Introduction

Convolutional neural networks (CNNs) have driven remarkable advances in visual recognition for the last decade. However, their performance is often degraded when training and test data are drawn from different distributions [8, 21, 46]. As such a distribution shift appears frequently and significantly in the wild, it has been a major obstacle to applying CNNs to real-world applications. The most common solution to this issue is domain adaptation [8, 30, 41, 42], which aims at adapting a model trained on source domains to a known target domain. However, domain adaptation models in general do not well generalize to unseen domains since they assume a single target domain.

Domain generalization (DG) [1, 2, 7, 22, 32] has been studied to resolve this limitation of domain adaptation. The goal of DG is to improve the generalization capability of a model on arbitrary domains unseen at training time. DG has been achieved by learning domain-invariant features [14, 24, 26, 39, 48] that capture semantics relevant to the target task while not being biased towards domain-specific characteristics. In this context, styles of images have been used to characterize their domains [27, 58]; it has been demonstrated that reducing model bias towards styles could improve the generalization ability [5, 33]. As a simple yet effective realization of this idea, style augmentation has been investigated recently [16, 17, 51, 58]. It allows a model to be unbiased to particular style distributions by augmenting training images with varying styles. Although they have been proven to be effective for domain generalization, however, there is still room for further improvement in terms of the style diversity; existing style augmentation methods obtain styles for augmentation from a restricted set of external images [17, 51] or by interpolating styles of source domain images [58], both of which lead to a limited range of styles.

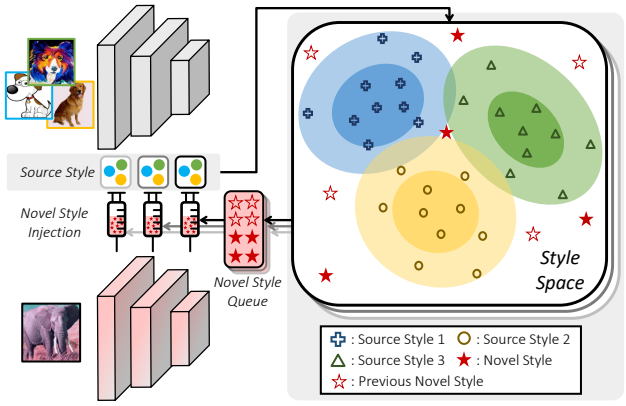

Figure 1. The motivation of our method. We improve the generalization ability of a model by adaptively synthesizing diverse, plausible, and novel styles that are distinct from both source domain styles and previously synthesized novel styles, then injecting them into intermediate features of the model during training for learning style-invariant representation.

In this paper, we present a novel framework to further enlarge the benefit of style augmentation. The key idea is to *constantly* generate novel and plausible styles and augment training images with the synthetic styles. Specifically, to

be novel, synthetic styles generated by our method should be distinct not only from styles of source domain images but also from previously generated synthetic styles, as illustrated in Fig. 1. To be plausible, on the other hand, they should not deviate too much from real image styles.

For efficient style synthesis, our framework begins with sampling a few prototypes that well represent the entire distribution of source image styles. Then the source style prototypes and previously synthesized novel styles are used to approximate the distribution of styles that have been observed by the model. To synthesize novel styles, we first generate plausible candidates of novel styles by jittering source image styles with random noises, and then sample a subset of such candidates that are diverse and not well represented by the approximate distribution of observed styles. This sampling process is implemented efficiently using (1) style queues that store source image styles and previously synthesized novel styles, and (2) score functions that measure the quality of sampled source style prototypes and novel styles. In particular, we employ monotone submodular score functions so that near-optimal prototypes and novel styles can be efficiently estimated by a greedy algorithm.

Our method is evaluated and compared with previous work on four public benchmarks for DG: PACS [21], OfficeHome [45], and DomainNet [35] for image classification, and the other for cross-domain person re-ID [28, 54]. Extensive experiments on these benchmarks demonstrate that our method is capable of achieving state-of-the-art domain generalization performance. The contribution of this paper is three-fold:

- We propose a novel approach to domain generalization that constantly synthesizes novel, diverse, and plausible styles to maximize the generalization effect of style augmentation.
- We present a novel framework based on style queues and submodular optimization for maintaining and generating styles effectively and efficiently.
- Our method outperforms existing DG techniques on four public benchmarks, in particular on those depicting large domain discrepancy.

2. Related work

Domain generalization. Domain generalization aims to generalize to the unseen domain by training with multiple source domains. Motivated by domain adaptation methods, initial studies for DG carried out domain alignment [14, 24, 26, 39, 48] to learn domain invariant features by reducing the distance of distribution among multiple domains. Specifically, most methods were implemented by adversarial learning [14, 26, 39], minimizing KL divergence [24, 48] and minimizing maximum mean discrepancy (MMD) [23]. In addition, self-supervised learning [3], ensemble learning [37, 50], and meta-learning [22] have been also studied on. Recent studies have focused on data augmentation [25, 56, 57] using a generator network. DDAIG [56] and PDEN [25] used domain adversarial learning to generate augmented data for multi-source and single-source domain generalization, respectively. L2A-OT [57] increased the generalization ability of a model by generating pseudo-novel domain images different from each source domain using a conditional generator. However, these methods have a problem of lack of diversity in the novel domains since optimization becomes more difficult when learning to synthesize more novel domains than the number of source domains. Our method is free from this limitation in that it allows the model to generate not images, but the novel styles at the feature level. In particular, it generates novel styles which have not been observed so far, then style augmentation with them allows the model to improve generalization by recognizing diverse novel styles.

Neural style representation. Neural style transfer has been focused on understanding the style information not relevant to the content. Gatys *et al.* [9] first studied that the style of an image can be captured by feature statistics of CNNs. In particular, AdaIN [12] showed that the channel-wise mean and standard deviation of features also can represent the style of an image. Recent studies [13, 17, 33, 58] utilize the style information as the characteristic of the domain, and they use the style representation at the feature level for domain generalization. MixStyle [58] mixed the feature statistics of source instances for simulating novel styles and injected them to regularize the CNN. However, they only consider a limited range of styles calculated by external images [17, 51] or formed by a linear interpolation of features statistics in the source domains [58]. Our method is free from this limitation since it synthesizes novel styles distinct from both the source styles and previously generated styles to increase the style diversity.

Maximum mean discrepancy. The maximum mean discrepancy [10] is a measure of the difference between two distributions. It is widely used to measure or minimize the divergence between distributions for generative adversarial learning [20] and improving interpretability of data distributions [15]. In the field of domain adaptation [29–31] and generalization [23], MMD has been applied to measure the divergence between different domains in the high-level space. MMD is utilized in our method as well, but to measure the discrepancy in the style space for selecting prototypes representing the source style distribution.

3. Method

Following previous work on DG [17, 33, 58], we assume that a domain can be characterized by the styles of its sam-

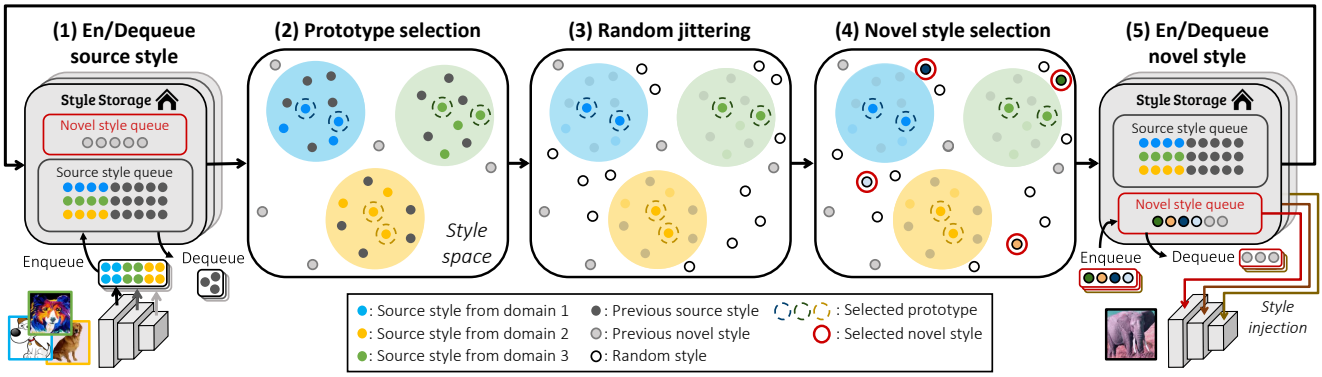

Figure 2. Overall pipeline of the proposed method. (1) For each iteration of training, source styles are computed from the source domain images by a network. Then, we enqueue them and dequeue previous source styles in the source style queues. (2) Source style prototypes that represent the style distributions of the source style queues are selected. (3) Candidates of novel styles are generated by jittering the source styles with random noises. (4) We select novel styles not represented by both prototypes of source domains and previous novel styles. (5) Selected novel styles are enqueued and previous novel styles are dequeued in the novel style queues. Then, randomly selected novel styles in the novel style queues are injected into our model during training on the fly. Steps (2)-(5) are executed every predefined number of iterations to constantly seek novel styles.

ples, and accordingly, that style-invariant representations will generalize well to arbitrary unseen domains. In this context, as a solution to DG, we propose a new framework for learning a style-invariant model via style augmentation. The key idea is to constantly feed a CNN with training data whose styles have not been observed before for maximizing the effect of style augmentation. To implement this idea, our framework constantly generates synthetic yet plausible styles that are distinct from those observed at the previous iterations, and replaces styles of training images with these synthetic ones while preserving semantic information of the images. The remainder of this section presents an overview of our framework (Sec. 3.1), the detailed algorithm for novel style synthesis (Sec. 3.2), and the training strategy with the novel styles (Sec. 3.3).

3.1. Overview

Our method represents the style of an image by channel-wise mean and standard deviation $\mu(\mathbf{Z}), \sigma(\mathbf{Z}) \in \mathbb{R}^C$ of its feature map $\mathbf{Z} \in \mathbb{R}^{C \times H \times W}$ [58] as follows:

$$\mu(\mathbf{Z}) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{Z}_{\cdot,h,w}, \quad (1)$$

$$\sigma(\mathbf{Z}) = \sqrt{\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} (\mathbf{Z}_{\cdot,h,w} - \mu(\mathbf{Z}))^2}, \quad (2)$$

where H and W denote height and width of the feature map.

In advance of synthesizing novel styles, we first approximate the distribution of styles that the network has observed so far. To approximate and track the style distribution, our method deploys two types of style queues: source style queues and novel style queues. The source style queues keep the styles of recently observed source images. On the other hand, the novel style queues store novel styles that are synthesized to be distinct from previously observed ones in both queues. Note that $\mu(\mathbf{Z})$ and $\sigma(\mathbf{Z})$ are kept separately. When the number of stored styles exceeds the limit, the styles are dequeued from the oldest. Based on these style queues, we constantly seek novel styles by an iterative procedure of selecting source prototypes and synthesizing novel styles, as presented in Fig. 2.

3.2. Novel style synthesis

We ensure that novel styles meet two criteria: diversity and plausibility. For the diversity, we seek styles not observed at previous iterations. At the same time, they should be plausible, *i.e.*, not too much deviated from the distribution of real source styles, in order to provide realistic styles.

To this end, we propose a novel style synthesis method composed of three steps: prototype selection, random jittering, and novel style selection. First, a few representatives of source styles, called source style prototypes, are selected to identify the source style distribution efficiently in a non-parametric way (Fig. 2(2)). Also, candidates of novel styles are generated by jittering the source styles with random noises (Fig. 2(3)). Then a subset of the candidates that are most distinct from the prototypes and previously generated novel styles are chosen as novel styles (Fig. 2(4)). By iterating these steps, novel styles different from what have been observed can be continually synthesized and stored in the novel style queues. The remaining part of this section elaborates on each step of novel style synthesis.

Prototype selection. We select m_p prototypes that well represent the distribution of source styles stored in the source style queue. Suppose that we have a set of source

styles $\mathcal{S}$ stored in the queue. Let $\mathcal{P} \subseteq \mathcal{S}$ be the prototype set. Inspired by a sampling technique for interpretable machine learning [15], we adopt the squared maximum mean discrepancy (MMD) between $\mathcal{S}$ and $\mathcal{P}$ with a kernel function k to measure the discrepancy between them:

$$\begin{aligned}\text{MMD}_k^2(\mathcal{S},\mathcal{P}) = &\frac{1}{|\mathcal{S}|^2}\sum_{\mathbf{s}_i,\mathbf{s}_j\in\mathcal{S}} k(\mathbf{s}_i,\mathbf{s}_j)\\ &-\frac{2}{|\mathcal{S}||\mathcal{P}|}\sum_{\mathbf{s}_i\in\mathcal{S},\mathbf{p}_j\in\mathcal{P}} k(\mathbf{s}_i,\mathbf{p}_j)\\ &+\frac{1}{|\mathcal{P}|^2}\sum_{\mathbf{p}_i,\mathbf{p}_j\in\mathcal{P}} k(\mathbf{p}_i,\mathbf{p}_j).\end{aligned} \tag{3}$$

To select the most representative styles $\mathcal{P}$ whose distribution is close to that of $\mathcal{S}$, the score function is designed as

$$\begin{aligned}J_b(\mathcal{P}) &= \frac{1}{|\mathcal{S}|^2}\sum_{\mathbf{s}_i,\mathbf{s}_j\in\mathcal{S}} k(\mathbf{s}_i,\mathbf{s}_j) - \text{MMD}_k^2(\mathcal{S},\mathcal{P})\\ &= \frac{2}{|\mathcal{S}||\mathcal{P}|}\sum_{\mathbf{s}_i\in\mathcal{S},\mathbf{p}_j\in\mathcal{P}} k(\mathbf{s}_i,\mathbf{p}_j) - \frac{1}{|\mathcal{P}|^2}\sum_{\mathbf{p}_i,\mathbf{p}_j\in\mathcal{P}} k(\mathbf{p}_i,\mathbf{p}_j),\end{aligned} \tag{4}$$

where the first constant term is introduced to guarantee that $J_b(\emptyset) = 0$, *i.e.*, J_b is a normalized score function. We select the prototypes $\mathcal{P}$ maximizing the objective:

$$\max_{\mathcal{P}\subseteq\mathcal{S},|\mathcal{P}|\leq m_p} J_b(\mathcal{P}). \tag{5}$$

While this maximization problem is generally known to be intractable, it has been proved that a greedy procedure returns a near-optimal solution for any normalized monotonic submodular function [34]. Since Eq. (4) with the radial basis function (RBF) kernel $k(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\gamma||\mathbf{x}_i - \mathbf{x}_j||)$ is monotonic and submodular as proven in [15], the prototype selection is done by the greedy forward selection, *i.e.*, repeatedly sampling the style that increases the score function the most as a prototype.

Random jittering for style candidates. As candidates for novel styles, a set of random styles $\mathcal{D}$ are generated by adding random noises to the source styles $\mathcal{S}$. First, we calculate the channel-wise standard deviation $\sigma(\mathcal{S}) \in \mathbb{R}^C$ of source styles $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, ...\mathbf{s}_N\}$. Random noise vectors are then sampled from a Gaussian distribution, $\mathcal{N}\big(0, \lambda \cdot \text{diag}(\sigma(\mathcal{S}))\big)$, where λ is a scalar hyper-parameter; the standard deviations of the Gaussian distribution is set proportional to $\sigma(\mathcal{S})$ for sampling plausible noises by considering real source style distributions. The sampled noises are then added to source styles for constituting $\mathcal{D}$ of diverse and plausible random styles. We then sample a fixed number of novel styles from $\mathcal{D}$ in the next step.

Novel style selection. To guarantee the diversity of the novel styles, we select m_c novel styles that are not well represented by an approximate distribution of observed styles. Let $\mathcal{P}' = \mathcal{P} \cup \mathcal{V}$ be the total set of observed styles where $\mathcal{V}$ is the set of previously synthesized styles in the novel style queue. To quantify the quality of a novel style, we adopt the following witness function:

$$g(\mathbf{x}) = \frac{1}{|\mathcal{D}|}\sum_{\mathbf{d}_i\in\mathcal{D}} k(\mathbf{x},\mathbf{d}_i) - \frac{1}{|\mathcal{P}'|}\sum_{\mathbf{p}_j\in\mathcal{P}'} k(\mathbf{x},\mathbf{p}_j), \tag{6}$$

where the first term measures the similarity to the novel style candidates, and the second term measures the similarity to the observed styles. A novel style that maximizes the witness function will well represent the novel style candidates and at the same time be distinct from the observed styles. The score function for sampled novel styles $\mathcal{C} \subseteq \mathcal{D}$ is then given by

$$L(\mathcal{C}) = \sum_{\mathbf{x}_l\in\mathcal{C}} g(\mathbf{x}_l). \tag{7}$$

Moreover, we additionally adopt the log-determinant regularizer [15] that encourages the diversity of selected novel styles in the process of the optimization and is known to be submodular [19]. The regularizer is formally given by

$$r(\mathcal{C}) = \log\det \mathbf{K}_{\mathcal{C},\mathcal{C}}, \tag{8}$$

where $\mathbf{K}_{\mathcal{C},\mathcal{C}}$ is the kernel matrix with entries $k_{i,j} = k(\mathbf{x}_i,\mathbf{x}_j)$ for all $\mathbf{x}_i,\mathbf{x}_j \in \mathcal{C}$. Finally, we select the novel styles maximizing the following score function:

$$\max_{\mathcal{C}\subseteq\mathcal{D},|\mathcal{C}|\leq m_c} L(\mathcal{C}) + r(\mathcal{C}). \tag{9}$$

Since the score function in Eq. (9) is monotone submodular, the optimization is done also by a greedy algorithm that chooses the novel style increasing the function the most repeatedly; the sampled novel styles are then stored in the novel style queue.

In summary, our novel style synthesis process consists of these 3 steps and is executed every predefined number of iterations to constantly seek novel styles in the entire learning process. The process is performed separately for the mean and standard deviation components to synthesize respective novel styles.

3.3. Training with novel styles

During training the target model, we diversify style of feature maps of input images by injecting the synthetic novel styles on the fly. Following previous work [12, 17, 58], we first normalize the feature maps by Instance Normalization [43] and then inject the novel styles into the style-normalized feature maps. For a feature map $\mathbf{Z} \in \mathbb{R}^{C\times H\times W}$, this style injection is formulated by

$$\text{StyIn}(\mathbf{Z};\mathbf{a},\mathbf{b}) = \mathbf{a}\cdot\frac{\mathbf{Z}-\mu(\mathbf{Z})}{\sigma(\mathbf{Z})} + \mathbf{b}, \tag{10}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{R}^C$ are random novel styles for standard deviation and mean, respectively. It can be applied to multiple convolutional blocks of the network, which is further discussed in Sec. 4.2. The remaining part of this section describes the overall training procedure and loss functions incorporating the novel style injection.

Let $f = f^{(2)} \circ f^{(1)}$ denote the target network, and suppose that novel styles are injected to the output of $f^{(1)}$. Given a source image $\mathbf{X}$ as input along with its one-hot label vector $\mathbf{y}$, the network is trained by minimizing the ordinary cross-entropy loss:

$$\mathcal{L}_{\text{ce}}^{\text{ori}} = -\mathbf{y}^\top \log f(\mathbf{X}). \tag{11}$$

Meanwhile, the source styles, $\mu(\mathbf{Z})$ and $\sigma(\mathbf{Z})$ where $\mathbf{Z} = f^{(1)}(\mathbf{X})$, are computed and stored in the source style queues, respectively. We then forward the same image to the network while injecting novel styles to its feature map $\mathbf{Z}$, and apply the cross-entropy loss to the output:

$$\mathcal{L}_{\text{ce}}^{\text{sty}} = -\mathbf{y}^\top \log f^{(2)}(\text{StyIn}(f^{(1)}(\mathbf{X}))). \tag{12}$$

Optimizing the two cross-entropy losses enables $f^{(2)}$ to be style-invariant and well-generalized. To further boost the generalization capability, we in addition introduce losses that force the consistency between the soften prediction for the original input and that for the style-injected ones. Specifically, the losses are formulated as the Kullback-Leibler (KL) divergence between the predictions:

$$\mathcal{L}_{\text{const}}^{\text{o2s}} = \text{KL}(f(\mathbf{X})/\tau \,||\, f^{(2)}(\text{StyIn}(f^{(1)}(\mathbf{X})))/\tau), \tag{13}$$
$$\mathcal{L}_{\text{const}}^{\text{s2o}} = \text{KL}(f^{(2)}(\text{StyIn}(f^{(1)}(\mathbf{X})))/\tau \,||\, f(\mathbf{X})/\tau), \tag{14}$$

where τ is a temperature hyper-parameter. Combining all together, the total objective is given by

$$\mathcal{L}_{\text{total}} = (1-w_1)\mathcal{L}_{\text{ce}}^{\text{ori}} + w_1\mathcal{L}_{\text{ce}}^{\text{sty}} + w_2(\mathcal{L}_{\text{const}}^{\text{o2s}} + \mathcal{L}_{\text{const}}^{\text{s2o}}), \tag{15}$$

where w_1 and w_2 are balancing hyper-parameters. In summary, the network is trained using the objective in Eq. (15), and meanwhile, source styles are stored in the queues. As training progresses, the novel style synthesis step is regularly executed to constantly seek novel styles.

4. Experiments

4.1. Datasets for evaluation

Generalization in image classification. The proposed method is evaluated on three conventional DG benchmarks for image classification. (1) **PACS** [21] consists of four domains, *i.e.*, Art Painting, Cartoon, Photo, and Sketch, and contains 9,991 images of 7 classes with large domain discrepancy. (2) **OfficeHome** [45] includes 15,500 images of 65 classes from four domains, Art, Clipart, Product, and Real World. (3) **DomainNet** [35] is a large-scale dataset that contains 586,575 images of 345 classes from six domains, Clipart, Infograph, Painting, Quickdraw, Real, and Sketch. For fair comparisons with previous work, we follow the leave-one-domain-out-protocol [49,56,57]. In detail, we choose one domain as the test domain and use the remaining domains as the source domains; the model showing the best performance on the validation splits of all source domains are chosen as the final model. The evaluation metric is the top-1 classification accuracy.

Generalization in instance retrieval. Our method is also evaluated for cross-domain person re-identification (re-ID) [56–58]. The goal of this task is to retrieve target person from multiple disjoint cameras, which are considered as different domains. We adopt the Market1501 [52] and DukeMTMC-reID (Duke) [36, 53] datasets. Market1501 consists of 32,668 images of 1,501 identities captured by 6 cameras and Duke contains 36,411 images of 1,812 identities captured by 8 cameras. Our model is trained on one dataset and tested on the other. In this task, the label space is disjoint between training and test identities. Mean Average Precision (mAP) and ranking accuracy are used for evaluation metrics.

4.2. Implementation details

Generalization in image classification. ResNet [11] pre-trained on ImageNet [6] is adopted as our classification network. The novel style injection is applied to the outputs of 1st and 2nd residual blocks of the network. For PACS and OfficeHome, our network is trained by SGD with batch size of 16 and weight decay of 5e-4 for 50 epochs and 25 epochs. The initial learning rate is set to 0.001 and decayed by 0.1 at 80% of total epochs. We adopt the augmentation strategy used in [3, 49]. For DomainNet, we use Adam optimizer [18] and inverse learning rate scheduling following [4], and train the network for 20 epochs. For all datasets, $\tau = 4$ and $w_2 = 2$ for the consistency regularization loss with sigmoid-rampup [40] at initial 5 epochs. Loss balancing weight w_1 is set to 0.1 for OfficeHome and 0.5 for the others. For the novel style synthesis, we set the synthesis cycle as 32 iterations. The length of the source style queue and that of the novel style queue are 1024 and 128, respectively. For DomainNet, the number of prototypes and that of novel styles in a single novel style synthesis are all 32, and for the other datasets, are 8 and 16, respectively.

Generalization in instance retrieval. Two different networks are adopted: ResNet50 and OSNet-IBN [55]. In both architectures, the style injection is applied to the outputs of the 1st and 2nd residual blocks. The re-ID model is trained for classification where each person identity is considered as a class. For fair comparisons with previous work, we use l_2 normalized features for OSNet-IBN and reproduce MixStyle [58] on the same setting by their public code.

Method	Art	Cartoon	Photo	Sketch	Avg.
ResNet18					
Baseline	77.63	76.77	95.85	69.50	79.94
MetaReg [1]	83.70	77.20	95.50	70.30	81.70
Jigen [3]	79.42	75.25	96.03	71.35	80.51
DDAIG [56]	84.20	78.10	95.30	74.70	83.10
L2A-OT [57]	83.30	78.20	**96.20**	73.60	82.80
EISNet [47]	81.89	76.44	95.93	74.33	82.15
SagNet [33]	83.58	77.66	95.47	76.30	83.25
MixStyle [58]	84.10	78.80	96.10	75.90	83.70
DSON [37]	84.67	77.65	95.87	82.23	85.11
FACT [49]	**85.37**	78.38	95.15	79.15	84.51
Ours	$84.41_{\pm 0.62}$	**79.25**$_{\pm 0.98}$	$94.93_{\pm 0.07}$	**83.27**$_{\pm 2.03}$	**85.47**
ResNet50					
Baseline	84.94	76.98	**97.64**	76.75	84.08
MetaReg [1]	87.20	79.20	97.60	70.30	83.60
EISNet [47]	86.64	81.53	97.11	79.07	85.84
DSON [37]	87.04	80.62	95.99	82.90	86.64
FACT [49]	89.63	81.77	96.75	84.46	88.15
Ours	**90.35**$_{\pm 0.62}$	**84.20**$_{\pm 1.43}$	$96.73_{\pm 0.46}$	**85.18**$_{\pm 0.46}$	**89.11**

Table 1. Leave-one-domain-out generalization results on PACS.

Method	Art	Clipart	Product	Real	Avg.
Baseline	57.88	52.72	73.57	74.80	64.72
MMD-AAE [23]	56.50	47.30	72.10	74.80	62.70
CrossGrad [38]	58.40	49.40	73.90	75.80	64.40
Jigen [3]	53.04	47.51	71.47	72.79	61.20
SagNet [33]	60.20	45.38	70.42	73.38	62.34
DDAIG [56]	59.20	52.30	74.60	76.00	65.50
MixStyle [58]	58.70	53.40	74.20	75.90	65.50
L2A-OT [57]	**60.60**	50.10	**74.80**	**77.00**	65.60
FACT [49]	60.34	54.85	74.48	76.55	**66.56**
Ours	$59.55_{\pm 0.21}$	**55.01**$_{\pm 0.29}$	$73.57_{\pm 0.28}$	$75.52_{\pm 0.21}$	65.89

Table 2. Leave-one-domain-out generalization results on Office-Home.

4.3. Quantitative results in image classification

Evaluation on PACS. Quantitative results of our and existing methods are summerized in Table 1; the baseline model is trained only with the cross-entropy loss. Our method consistently achieves the best performance in the averaged accuracy regardless of the type of its backbone network. In detail, ours outperforms existing methods in three test domains (Art, Cartoon, and Sketch) with ResNet50. When incorporating ResNet18, our method clearly surpasses MixStyle [58], a style augmentation technique based on linear interpolation of known styles. Unlike previous work synthesizing novel domain samples, such as L2A-OT [57] and DDAIG [56], our method requires neither data generator nor domain label. The only overhead of ours is the memory footprint for the style queues, which is mostly negligible. While being simpler and imposing less overhead, ours improves performance substantially since it enables to learn style-invariant representation effectively by synthesizing novel and diverse styles on the fly. Overall, these results demonstrate the efficacy of our method for domain generalization and justifies our motivation of constantly seeking diverse novel styles to prevent the style bias.

Method	Clip.	Info.	Paint.	Quick.	Real	Sketch	Avg.
ResNet18							
Baseline	56.56	18.44	45.30	12.47	57.90	38.83	38.25
MetaReg [1]	53.68	**21.06**	45.29	10.63	**58.47**	42.31	38.57
DMG [4]	60.07	18.76	44.53	14.16	54.72	41.73	39.00
Our	**60.14**$_{\pm 0.48}$	$17.82_{\pm 0.32}$	**46.52**$_{\pm 0.23}$	**14.58**$_{\pm 0.15}$	$55.36_{\pm 0.98}$	**45.26**$_{\pm 0.53}$	39.95
ResNet50							
Baseline	64.04	23.63	51.04	13.11	64.45	47.75	44.00
MetaReg [1]	59.77	**25.58**	50.19	11.52	**64.56**	50.09	43.62
DMG [4]	65.24	22.15	50.03	**15.68**	59.63	49.02	43.63
Our	**66.11**$_{\pm 0.66}$	$21.42_{\pm 0.12}$	**51.36**$_{\pm 0.37}$	$15.25_{\pm 0.35}$	$61.73_{\pm 0.23}$	**51.76**$_{\pm 0.21}$	44.60

Table 3. Leave-one-domain-out generalization results on DomainNet.

Method	Market1501 → Duke				Duke → Market1501			
	mAP	R1	R5	R10	mAP	R1	R5	R10
ResNet50								
Baseline	19.3	35.4	50.3	56.4	20.4	45.2	63.6	70.9
MixStyle [58]	23.8	42.2	58.8	64.8	24.1	51.5	69.4	76.2
Ours	**26.3**	**46.5**	**62.4**	**68.0**	**27.2**	**55.0**	**73.9**	**85.5**
OSNet-IBN								
Baseline	26.7	48.5	62.3	67.4	26.1	57.7	73.7	80.0
CrossGrad [38]	27.1	48.5	63.5	69.5	26.3	56.7	73.5	79.5
MixStyle* [58]	27.7	48.4	62.7	72.1	28.8	59.7	76.7	82.7
DDAIG [56]	28.6	**50.6**	65.2	70.3	29.0	60.9	77.1	83.2
L2A-OT [57]	29.2	50.1	64.5	70.1	30.2	63.8	**80.2**	84.6
Ours	**29.7**	**50.6**	**65.4**	**74.2**	**32.2**	**64.7**	**80.2**	**89.1**

Table 4. Generalization results on cross-domain person re-ID. (∗: Reproduced by the official implementation).

Evaluation on OfficeHome. OfficeHome is composed of four domains with less domain discrepancy compared with the other datasets. As summarized in Table 2, despite the small domain gap in this benchmark, which is unfavorable for domain generalization by synthesizing novel styles, our method is on par with the state of the art. In particular, our method consistently improves the performance of the baseline in all domains while most of existing methods perform poorly in certain domains. Note that ours also outperforms the methods synthesizing novel domain samples like L2A-OT [57] and MixStyle [58] in the averaged accuracy.

Evaluation on DomainNet. Table 3 presents the results on DomainNet consisting of 6 domains with much larger discrepancy than the other datasets. On this more challenging benchmark, our method shows better performance in the averaged accuracy than existing methods and improves the top-1 averaged accuracy by 1.70%p and 0.60%p using ResNet18 and ResNet50 backbones, respectively. While our method clearly outperform the baseline, existing methods are often inferior to the baseline when using ResNet50 backbone. Also, on both PACS and DomainNet, our method consistently improves the performance for both ResNet18 and ResNet50 models, respectively.

Style injection	Novel style synthesis	$\mathcal{L}^*_{\text{const}}$	Art	Cartoon	Photo	Sketch	Avg.
✗	✗	✗	77.63	76.77	95.85	69.50	79.94
✓	✗	✗	81.88	78.03	94.67	78.93	83.38
✓	✓	✗	84.38	78.21	95.05	80.13	84.44
✓	✓	✓	84.41	79.25	94.93	83.27	85.47

Table 5. Ablation studies on each component of our method on PACS with ResNet18. * denotes both s2o and o2s.

4.4. Quantitative results in instance retrieval

The efficacy of our method is also demonstrated on the person re-ID task. As summarized in Table 4, ours consistently improves performance in both of the two cross-domain scenarios, from Market1501 to Duke and vice versa. Our method is effective when using OSNet-IBN as well as ResNet. Even in the setting where each camera view is considered as a domain, our method surpasses previous work on DG in both mAP and ranking accuracy. These results demonstrate the superiority of our method over previous work synthesizing novel samples.

4.5. Ablation studies

Impact of each component. In Table 5, we carry out an ablation study to investigate the effect of each component: style injection, novel style synthesis, and consistency losses. The style injection of adding random noises to the original feature statistics improves the overall accuracy. Although the simple augmentation strategy that perturbs feature statistics with Gaussian noises is useful for DG, it is still inferior to existing methods. The performance is boosted and becomes comparable to the state of the art by injecting novel styles synthesized by our method instead of the random noises. This result validates the effectiveness of our novel style synthesis technique for style augmentation; it surpasses existing methods synthesizing novel domain samples such as L2A-OT [57] and MixStyle [58]. Lastly, the consistency losses further improve the performance, which enables our method to clearly outperform existing DG methods.

Diversity of novel styles. Our method synthesizes novel styles that are distinct from not only source styles but also those synthesized previously, which guarantees their diversity. In Fig. 3, we demonstrate the diversity by comparing synthetic styles generated by our method with those made by MixStyle [58], which is a representative style augmentation method for DG. First, we measure their diversity through their channel-wise deviations in Fig. 3(a), showing that synthetic styles of our method are substantially more scattered. Second, to examine how much the synthetic styles are distinct from source styles, we estimate the squared MMD between them; Fig. 3(b) suggests that ours generates styles more distinct from source styles.

Sensitivity to queue length. Our method introduces source

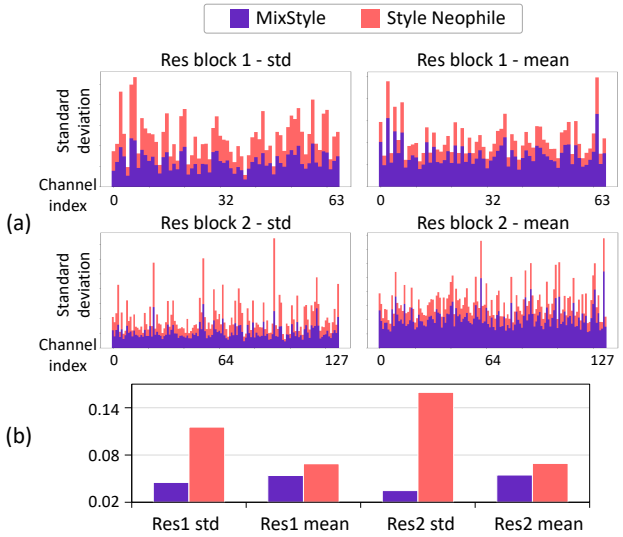

Figure 3. Empirical analysis on the diversity of styles synthesized by MixStyle and ours with ResNet18 on PACS. (a) Channel-wise deviations of synthesized styles. (b) MMD^2_k between source and synthesized styles.

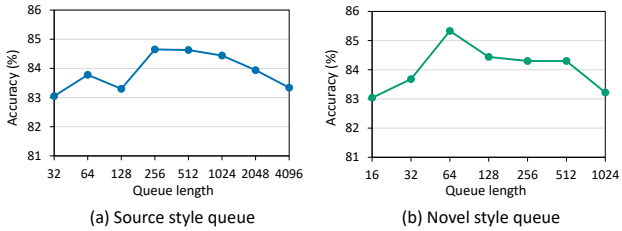

Figure 4. Evaluation on PACS with the change of queue length.

style queues and novel style queues to approximate the distribution of observed styles and thus is affected by the lengths of the queues to some extent. We investigate how sensitive our method to the length of each queue; the experiment is conducted without the consistency losses to clearly identify the effect of the queue length. Fig. 4 shows the results in the averaged top-1 accuracy measured by varying the length of each style queue on PACS using ResNet18. The performance is fairly high and stable in the length from 256 (64) to 1024 (512) of source (novel) style queues. Hence, we argue that our method is insensitive to the length of each style queue. Note that, in this experiment, we follow the hyper-parameter setting of our final model as-is; the setting is not optimal for this experiments, but our results in the setting still outperform or are comparable to those of existing methods.

Qualitative analysis of novel styles. To verify that our synthesized styles are novel, diverse, and plausible, we analyze and compare the synthetic novel styles with source styles in qualitative manners. Fig. 5(a) presents t-SNE [44] visualization of source and novel style distributions in the middle of training. It shows that novel styles are almost evenly

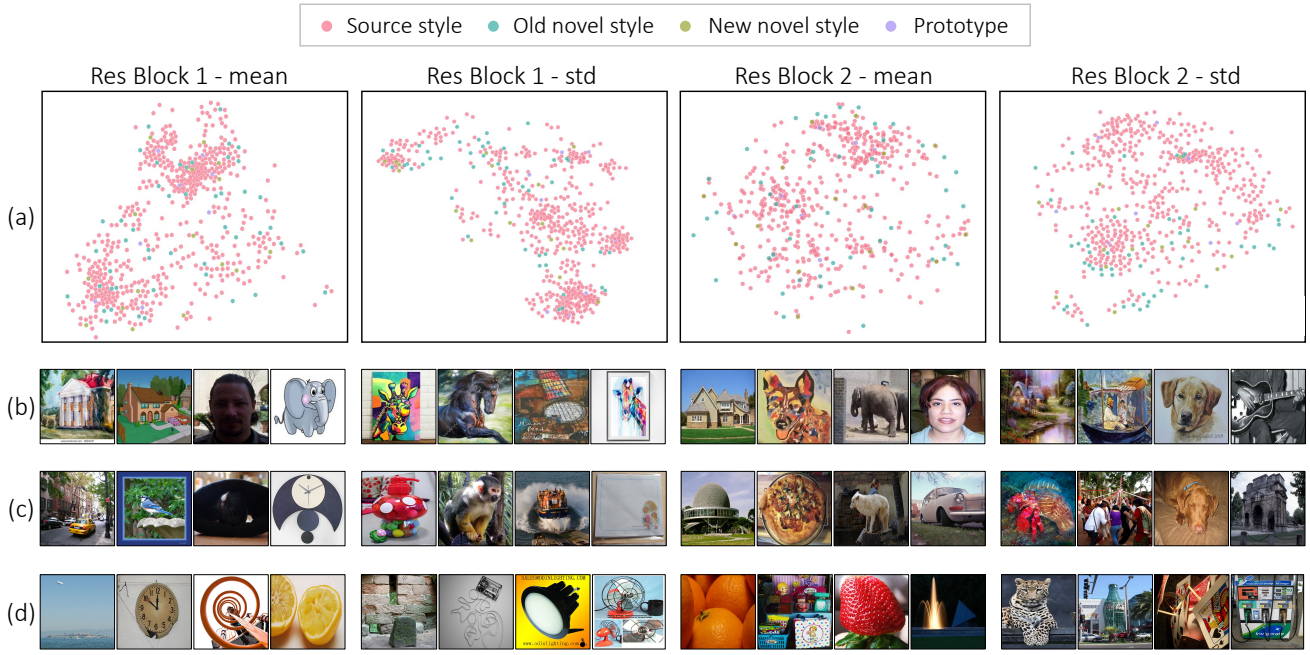

Figure 5. (a) t-SNE visualization of style vectors. The mean and standard deviation are computed from feature maps of the 1st and 2nd residual blocks of ResNet18 while being trained on PACS. (b) Examples of PACS images of source style prototypes. (c) Examples of ImageNet [6] images whose styles are closest to source style prototypes (b) in style space. (d) Examples of ImageNet images whose styles are closest to novel styles in the style space.

scattered (*i.e.*, diverse) and often occupy areas where source styles are not well distributed (*i.e.*, novel, as intended) while being not too much deviated from the distribution of source styles (*i.e.*, plausible). These properties of novel styles are also verified in another qualitative way through ImageNet examples whose styles are closest to the source and novel styles in the style spaces. Note that we utilize ImageNet examples for the analysis since our method does not generate images but directly synthesizes styles whose visualization is not straightforward. First, Fig. 5(b) and Fig. 5(c) show PACS examples of the source style prototypes and ImageNet examples closest to the prototypes, respectively, demonstrating the high similarity between them in terms of styles. In contrast, ImageNet examples in Fig. 5(d), whose styles are the closest to the novel styles, are diverse among themselves and show a larger discrepancy from those in Fig. 5(b) and Fig. 5(c).

5. Limitations

Our method has two limitations. First, it shows a large variance of performance in specific settings. Since our method performs the stochastic process in the random jittering step for novel style candidates, it causes the problem. Second, our method performs on-par on Office-Home, in which the discrepancy between domains is much smaller than the other datasets. When the discrepancy between domains is minimal, our strategy provides a relatively marginal performance boost since the influence of the novel style is diminished. In future work, we will improve the generalization ability while resolving the two problems.

6. Conclusion

We have proposed a novel method to learn style-invariant representations for domain generalization. It continually seeks novel, diverse, and plausible styles to maximize the benefit of style augmentation. Based on the two types of style queues, we efficiently approximate the style distribution that has been observed so far and generate novel styles that are different from observed styles including both source and previously synthesized novel styles. Since the process is formulated as monotone submodular optimization tasks, it can be conducted by greedy algorithms. Then, we inject synthesized novel styles into the feature map, which can reduce model bias toward styles and increase the generalization ability. We confirm that our attempt to constantly seek and utilize novel styles is effective in domain generalization on multiple public benchmarks.

Acknowledgement. This work was supported by Samsung Research Funding & Incubation Center of Samsung Electronics under Project Number SRFC-IT1801-05 and Samsung Electronics Co., Ltd (IO201210-07948-01).

References

[1] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2018. 1, 6

[2] Gilles Blanchard, Gyemin Lee, and Clayton Scott. Generalizing from several related classification tasks to a new unlabeled sample. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2011. 1

[3] Fabio M Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. Domain generalization by solving jigsaw puzzles. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2, 5, 6

[4] Prithvijit Chattopadhyay, Yogesh Balaji, and Judy Hoffman. Learning to balance specificity and invariance for in and out of domain generalization. In *European Conference on Computer Vision*, pages 301–318. Springer, 2020. 5, 6

[5] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne T Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: a large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 5, 8

[7] Qi Dou, Daniel Coelho de Castro, Konstantinos Kamnitsas, and Ben Glocker. Domain generalization via model-agnostic learning of semantic features. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2019. 1

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proc. International Conference on Machine Learning (ICML)*, 2015. 1

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[10] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2006. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 5

[12] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 4

[13] Seogkyu Jeon, Kibeom Hong, Pilhyeon Lee, Jewook Lee, and Hyeran Byun. Feature stylization and domain-aware contrastive learning for domain generalization. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 22–31, 2021. 2

[14] Yunpei Jia, Jie Zhang, Shiguang Shan, and Xilin Chen. Single-side domain generalization for face anti-spoofing. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[15] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2016. 2, 4

[16] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12975–12984, 2020. 1

[17] Namyup Kim, Taeyoung Son, Cuiling Lan, Wenjun Zeng, and Suha Kwak. Wedge: Web-image assisted domain generalization for semantic segmentation. *arXiv preprint arXiv:2109.14196*, 2021. 1, 2, 4

[18] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2015. 5

[19] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(2), 2008. 4

[20] Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *arXiv preprint arXiv:1705.08584*, 2017. 2

[21] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 5542–5550, 2017. 1, 2, 5

[22] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 1, 2

[23] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 6

[24] Haoliang Li, YuFei Wang, Renjie Wan, Shiqi Wang, Tie-Qiang Li, and Alex C Kot. Domain generalization for medical imaging classification with linear-dependency regularization. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2

[25] Lei Li, Ke Gao, Juan Cao, Ziyao Huang, Yepeng Weng, Xiaoyue Mi, Zhengze Yu, Xiaoya Li, and Boyang Xia. Progressive domain expansion network for single domain generalization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[26] Ya Li, Xinmei Tian, Mingming Gong, Yajing Liu, Tongliang Liu, Kun Zhang, and Dacheng Tao. Deep domain generalization via conditional invariant adversarial networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2018. 1, 2

[27] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. *arXiv preprint arXiv:1701.01036*, 2017. 1

[28] Jiawei Liu, Zheng-Jun Zha, Di Chen, Richang Hong, and Meng Wang. Adaptive transfer network for cross-domain

person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7202–7211, 2019. 2

[29] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *Proc. International Conference on Machine Learning (ICML)*, 2015. 2

[30] Mingsheng Long, Jianmin Wang, Guiguang Ding, Jiaguang Sun, and Philip S Yu. Transfer joint matching for unsupervised domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2

[31] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2016. 2

[32] Krikamol Muandet, David Balduzzi, and Bernhard Schölkopf. Domain generalization via invariant feature representation. In *Proc. International Conference on Machine Learning (ICML)*, 2013. 1

[33] Hyeonseob Nam, HyunJae Lee, Jongchan Park, Wonjun Yoon, and Donggeun Yoo. Reducing domain gap via style-agnostic networks. *arXiv preprint arXiv:1910.11645*, 2(7):8, 2019. 1, 2, 6

[34] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978. 4

[35] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019. 2, 5

[36] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *Proc. European Conference on Computer Vision (ECCV)*, 2016. 5

[37] Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2, 6

[38] Shiv Shankar, Vihari Piratla, Soumen Chakrabarti, Siddhartha Chaudhuri, Preethi Jyothi, and Sunita Sarawagi. Generalizing across domains via cross-gradient training. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 6

[39] Rui Shao, Xiangyuan Lan, Jiawei Li, and Pong C Yuen. Multi-adversarial discriminative deep domain generalization for face presentation attack detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[40] Antti Tarvainen and Harri Valpola. Weight-averaged, consistency targets improve semi-supervised deep learning results. *CoRR,, vol. abs/1703*, 2017, 1780. 5

[41] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[42] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1

[43] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 4

[44] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 2008. 7

[45] Hemanth Venkateswara, Jose Eusebio, Shayok Chakraborty, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 5

[46] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 1

[47] Shujun Wang, Lequan Yu, Caizi Li, Chi-Wing Fu, and Pheng-Ann Heng. Learning from extrinsic and intrinsic supervisions for domain generalization. In *European Conference on Computer Vision*, pages 159–176. Springer, 2020. 6

[48] Ziqi Wang, Marco Loog, and Jan van Gemert. Respecting domain relations: Hypothesis invariance for domain generalization. In *Proc. International Conference on Pattern Recognition (ICPR)*, 2021. 1, 2

[49] Qinwei Xu, Ruipeng Zhang, Ya Zhang, Yanfeng Wang, and Qi Tian. A fourier-based framework for domain generalization. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14383–14392, 2021. 5, 6

[50] Zheng Xu, Wen Li, Li Niu, and Dong Xu. Exploiting low-rank structure from latent domains for domain generalization. In *Proc. European Conference on Computer Vision (ECCV)*, 2014. 2

[51] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2100–2110, 2019. 1, 2

[52] Liang Zheng, Liyue Shen, Lu Tian, Shengjin Wang, Jingdong Wang, and Qi Tian. Scalable person re-identification: A benchmark. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015. 5

[53] Zhedong Zheng, Liang Zheng, and Yi Yang. Unlabeled samples generated by gan improve the person re-identification baseline in vitro. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. 5

[54] Zhun Zhong, Liang Zheng, Zhiming Luo, Shaozi Li, and Yi Yang. Invariance matters: Exemplar memory for domain adaptive person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 598–607, 2019. 2

[55] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Omni-scale feature learning for person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3702–3712, 2019. 5

[56] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Deep domain-adversarial image generation for domain generalisation. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 13025–13032, 2020. 2, 5, 6

[57] Kaiyang Zhou, Yongxin Yang, Timothy Hospedales, and Tao Xiang. Learning to generate novel domains for domain generalization. In *Proc. European Conference on Computer Vision (ECCV)*, pages 561–578. Springer, 2020. 2, 5, 6, 7

[58] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 3, 4, 5, 6, 7